NEW DECOMPOSITION-BASED TECHNIQUES FOR SOLVING TWO-STAGE STOCHASTIC PROGRAMS: NETWORK INTERDICTION PROBLEMS

D. Morton (UT Austin)

J. Salmerón, K. Wood (NPS)

October 2000

The Problem

$$(P)\ \min_{x \in X}\ z = E\{f(\boldsymbol{x}, \tilde{\xi})\}, \quad \text{where}$$

$$f(\boldsymbol{x}, \tilde{\xi}) = c'\boldsymbol{x} + \min_{y \geq 0}\ \tilde{q}'\boldsymbol{y}$$

$$\text{s.t.}\quad \tilde{D}\,\boldsymbol{y} \leq \tilde{B}\,\boldsymbol{x} + \tilde{d}$$

x = First-stage decisions (before $\tilde{\xi}$ is known)

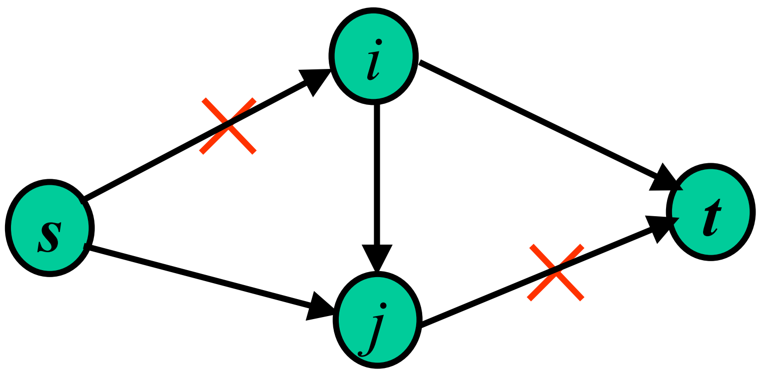
$y = y(x, \tilde{\xi})$ = Second-stage decisions $\tilde{\xi} = (\tilde{d}, \tilde{q}, \tilde{B}, \tilde{D})$

Network design (demand, transportation times)

Electric Power Generation (demand, generators availability, water inflows, spot market costs)

Network interdiction (attack successes, network data)

Network Interdiction Problems



MAX E\{ Min Length from s to t \}

MIN E\{ Max Flow from s to t \}

x_{ij} = 1 if interdiction of arc (i,j) is attempted, 0 otherwise

l_{ij}, d_{ij}= Nominal Arc (i,j) Length, Delay (*Shortest Path problem*)

u_{ij}= Nominal Arc (i,j) Capacity (*Maximum Flow problem*)

r_{ij}= Amount of resource needed to attempt to interdict the Arc (i,j)

ξ_{ij}= Attack success for Arc (i,j) (Random variable):

$$\tilde{l}_{ij} = l_{ij} + \xi_{ij} d_{ij} x_{ij}$$ (“Delay” for the *Shortest Path problem*)

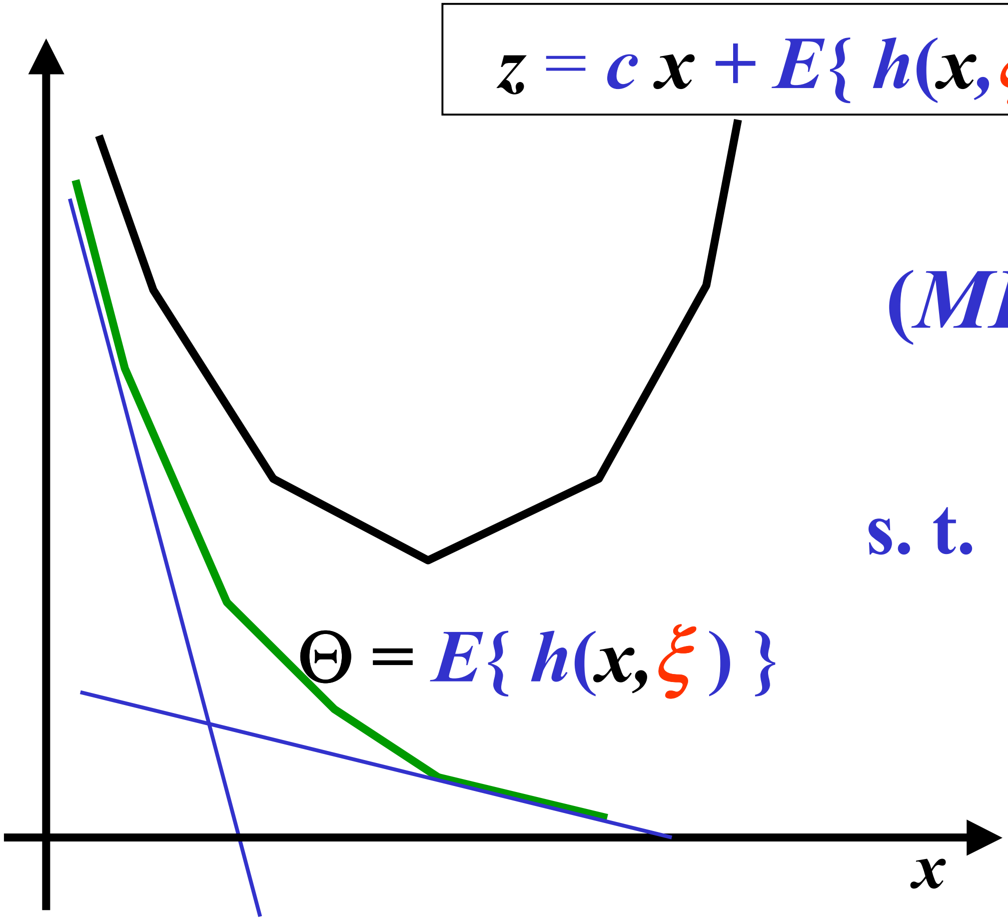
$$\tilde{u}_{ij} = u_{ij}(1 - \xi_{ij} x_{ij})$$ (“Diminished” capacity for the *Max. Flow Prob.*)

Our Approach:

Sampling Version of Benders Decomp

- Other researchers have worked in this arena, e.g., Higle and Sen, Dantzig and Glynn, Dantzig and Infanger
- Our approach is new, and probably conceptually simpler

Benders Decomposition (I)



$$(MP^K):\quad \min_{x \in X}\; c'x + \Theta$$

$$\text{s. t.}\quad \Theta \geq G_k\, x + g_k,\ \forall k \in K$$

Benders Decomposition (II)

Subproblem (and its dual) associated to a first stage feasible solution

$$SP(\hat{x}_k) : \min_{y \geq 0} \sum_{\omega \in \Omega} p^{\omega} q^{\omega} \boldsymbol{y}_k^{\omega}$$

$$\text{s.t.} \quad D^{\omega} \boldsymbol{y}_k^{\omega} \leq B^{\omega} \hat{x}_k + d^{\omega}, \forall \omega \in \Omega, \ (\boldsymbol{\pi}_k^{\omega})$$

This is a separable problem:

$$\forall \omega \in \Omega \qquad SP^{\omega}(\hat{x}_k) : \min_{y \geq 0} \ p^{\omega} q^{\omega} \boldsymbol{y}_k^{\omega}$$

$$\text{s.t.} \quad D^{\omega} \boldsymbol{y}_k^{\omega} \leq B^{\omega} \hat{x}_k + d^{\omega}, \ (\boldsymbol{\pi}_k^{\omega})$$

Benders Decomposition (III)

G and g are computed as the expectation of πB and πd

$$\boxed{\begin{aligned}\tilde{G}_k &= \tilde{\pi}_k \tilde{B} \\ \tilde{g}_k &= \tilde{\pi}_k \tilde{d}\end{aligned}}$$

$$\boxed{\begin{aligned}G_k &= E\left\{\tilde{G}_k\right\} = \int_\Omega \tilde{\pi}_k \tilde{B} \; \mathrm{d}\boldsymbol{P}(\omega) \\ g_k &= E\left\{\tilde{g}_k\right\} = \int_\Omega \tilde{\pi}_k \tilde{d} \; \mathrm{d}\boldsymbol{P}(\omega)\end{aligned}}$$

But exact values for G_k and g_k are unobtainable if

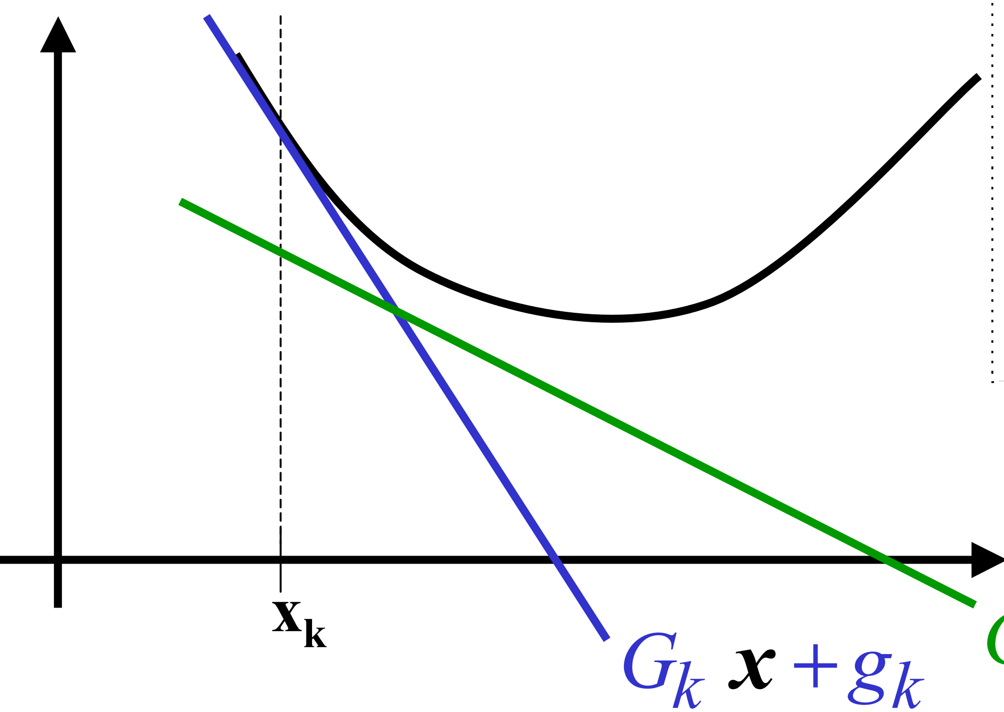
- The number of "scenarios" is large (even if finite)
- Some of the distributions are continuous

Benders Decomposition (IV)

"May we replace the actual G_k and g_k by estimators?"

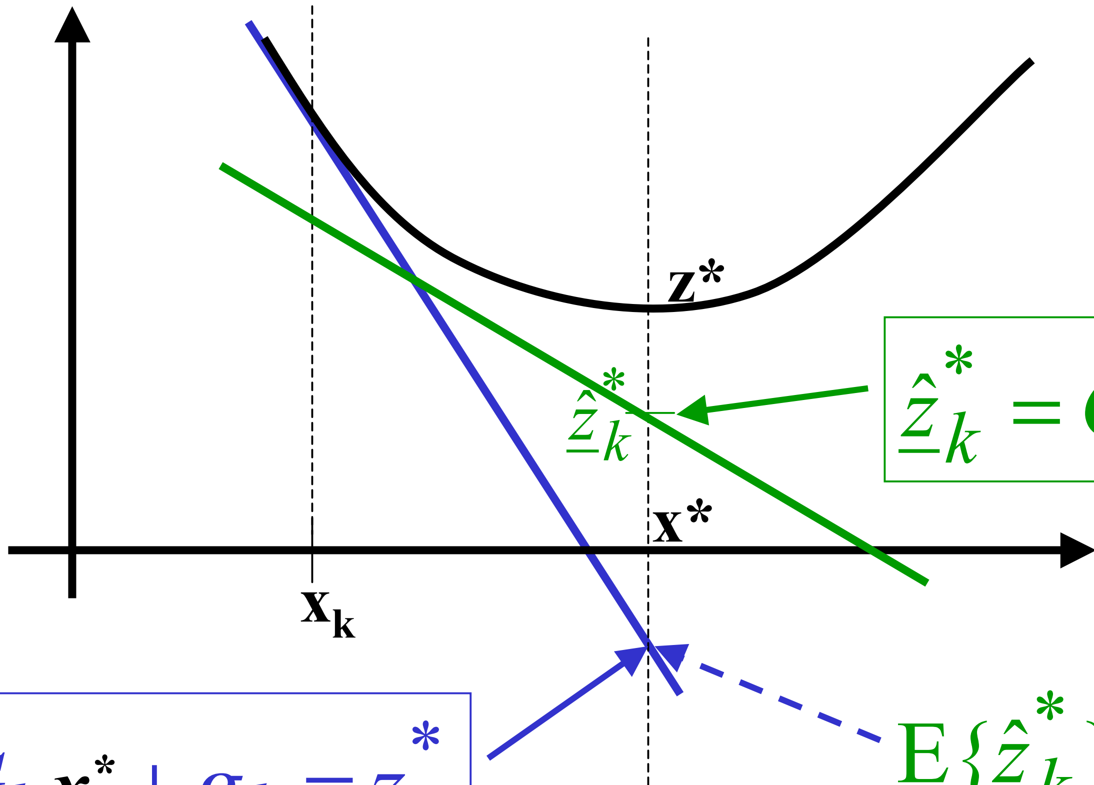
$$\hat{G}_k = \frac{1}{|S_k|} \sum_{\omega \in S_k} \pi_k^{\omega} B^{\omega}$$

$$\hat{g}_k = \frac{1}{|S_k|} \sum_{\omega \in S_k} \pi_k^{\omega} d^{\omega}$$



Estimation Procedure (I)

How do the estimated cuts behave at the optimal solution?



$$\hat{\underline{z}}_k^* = \hat{G}_k \boldsymbol{x}^* + \hat{g}_k$$

$$\mathrm{E}\{\hat{\underline{z}}_k^*\} = \underline{z}_k^*$$

$$G_k \boldsymbol{x}^* + g_k = \underline{z}_k^*$$

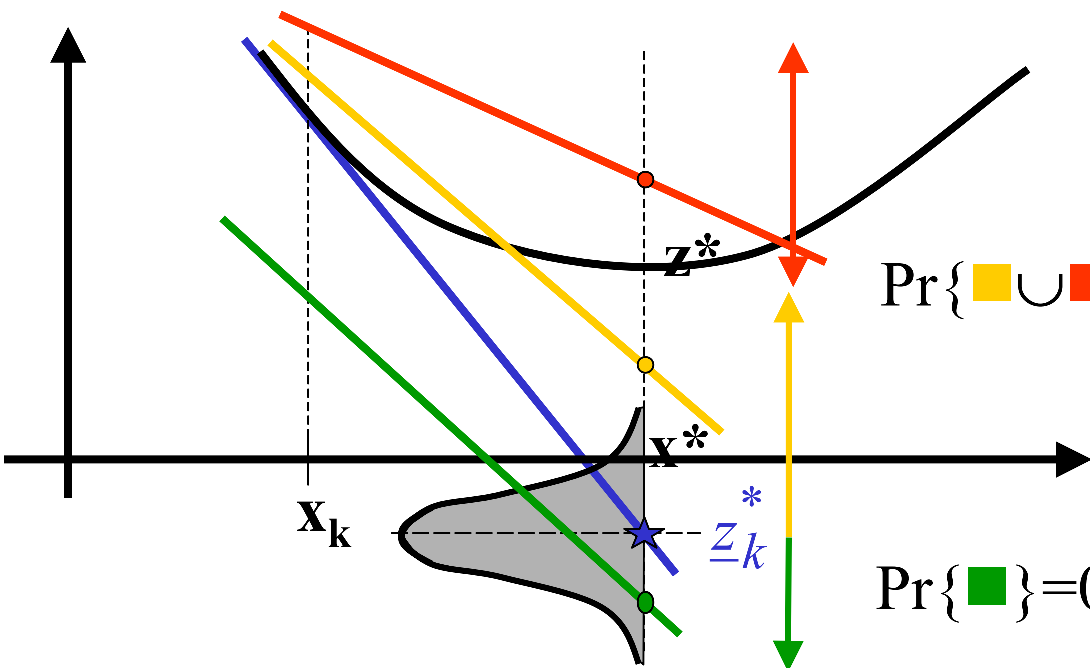
Estimation Procedure (II)

Hypothesis (C.L.T.)

$$\boxed{\left(\hat{G}_k, \hat{g}_k\right) \cong N_{n+1}\left(\begin{pmatrix} G_k \\ g_k \end{pmatrix} \begin{pmatrix} \Sigma_G & \Sigma_{G,g} \\ \Sigma_{G,g} & \sigma_g \end{pmatrix}\right)}$$

Thus

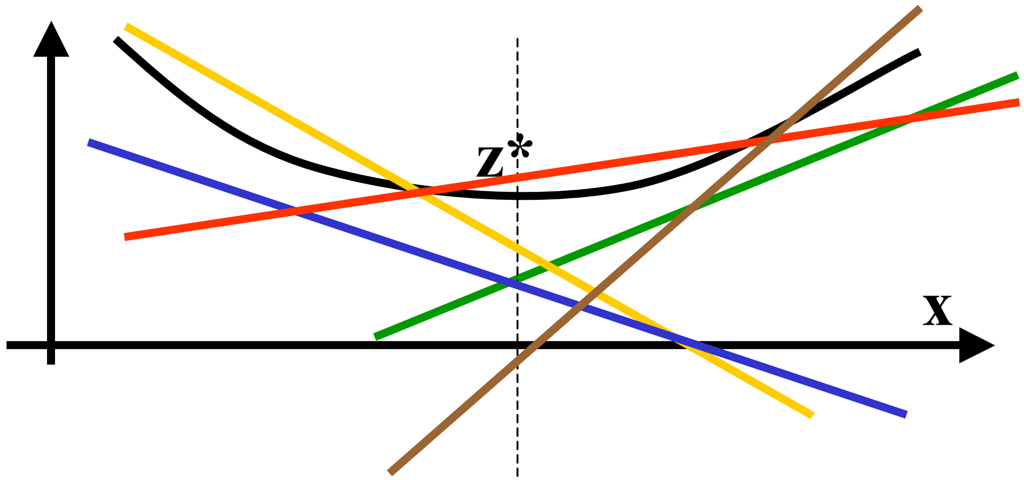
$$\boxed{\hat{\underline{z}}_k^* = \hat{G}_k \boldsymbol{x}^* + \hat{g}_k \equiv N_1(\underline{z}_k^*, \Lambda)}$$



Pr{"Success"}=

Pr{■∪■} ≥ 0.5

Estimation Procedure (III)



$\boldsymbol{p_n}$ = Probability that {at least $\boldsymbol{n}$ cuts (among **5**) are "valid" at $\boldsymbol{x^*}$}

$$\boxed{p_n \geq \sum_{m=n}^{5} \binom{5}{m} 0.5^5}$$

$$\rightarrow \begin{cases} \boldsymbol{p_5} = \Pr\{\text{All}\} \geq 3.1\% \\ \boldsymbol{p_4} \geq 18.7\ \% \\ \boldsymbol{p_3} \geq 50\% \\ \boldsymbol{p_2} \geq 81.2\% \\ \boldsymbol{p_1} \geq 96.8\% \end{cases}$$

In general, for a total of $\boldsymbol{n}$ cuts we may find $\boldsymbol{m=m(n, \alpha)}$ such that:

Pr{ at least $\boldsymbol{m}$ among $\boldsymbol{n}$ cuts are valid at $\boldsymbol{x^*}$ } $\geq$ $1-\alpha$

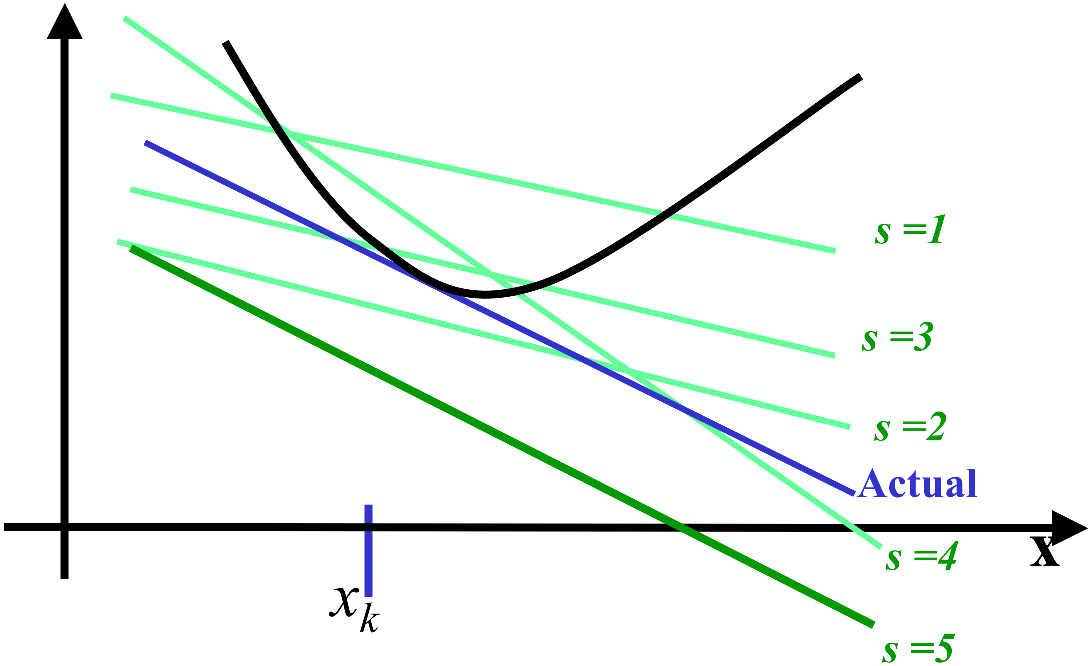
Probabilistic Bound (I)

Let s denote an index for multiple cuts at the same x_k

"Group of $s=1, 2, \ldots, n_k$ cuts at iteration k"

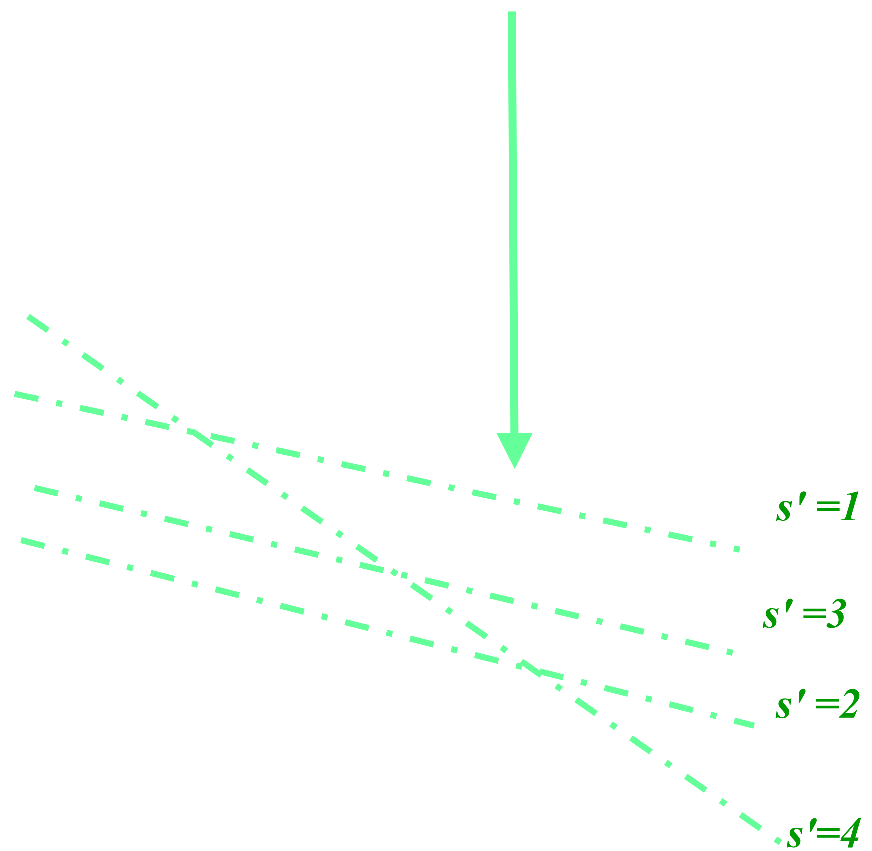
n_k **is the** $k\text{-}th$ **"group size"**

$(\hat{G}_k^s, \hat{g}_k^s)$



Pr { (at least) *one* of the 5 cuts is valid at x^* } = 0.968

<u>The "Weakest" Cut from each group</u>


s' =1
s' =3
s' =2
s'=4

Probabilistic Bound (II)

$$MP^K: \quad \min_{\Theta,\, \delta,\, x \in X} \quad c'x + \Theta$$

$$\text{s.t.} \begin{cases} \Theta \geq \hat{G}_k^s x + \hat{g}_k^s + (\delta_k^s - 1) M_k^s, \\ \qquad\qquad\qquad\qquad \forall k \in K,\, s = 1, \ldots, n_k \\ \displaystyle\sum_{s=1}^{n_k} \delta_k^s = 1,\ \forall k \in K; \quad \delta \in \{0,1\}^N \end{cases}$$

A_k={At least *one* cut among n_k in the *k-th* group lies below $\underline{z}_k^*$}

$$\Pr\{LB(K) \leq z^*\} \geq \Pr\{\bigcap_{k \in K} A_k\} = \prod_{k \in K} (1 - 0.5^{n_k})$$

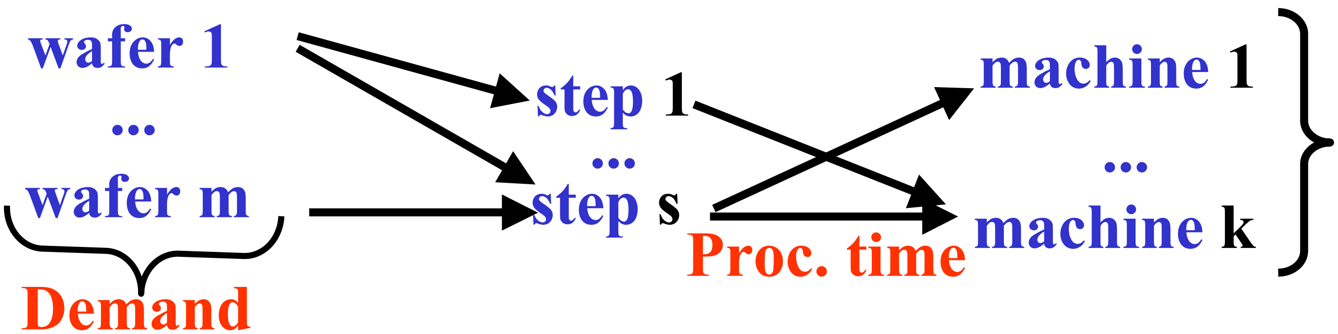
n_k =10 **for all the groups guarantees Prob > *0.95* for *50* iters.**

$n_k \approx 7.8 + 1.57 \log(k)$ **(*8, 9, 10, 11, 11, 11, 11, 12, 12,…*)**

guarantees Prob > *0.95* indefinitely

Computational Results (I)

SEMICONDUCTOR WAFER PRODUCTION-FACILITY EXPANSION



x_k = # mach. type k to be purchased (limited budget)

27 machine types (budget allows to buy 6 machines)

10 wafers: 5 scenarios of demand per wafer

7 steps: 2 scenarios per step-machine = $E\,(T_{sm})\cdot[\,1 \pm \alpha\,]$

From Morton and Wood (1999) Op. Res. 47, No. 6

α	x	Existing (LB,UB)	New (LB*,UB)
0.00	CONT.	(114.6, -)	(137.3, 145.8)
	INTEGER	(168.1, 179.9)	(134.9, 179.3)
0.10	CONT.	(34.24, -)	(138.6, 146.9)
	INTEGER	(86.6, 173.6)	(135.8, 171.6)
0.25	CONT.	(0.00, -)	(126.3, 131.7)
	INTEGER	(0.00, 153.4)	(125.3, 149.4)

(*) Prob. > 0.95 in all cases

Computational Results (II)

NETWORK CAPACITY EXPANSION

- x_k= How much capacity should be added to each arc **k** in a communications network (limited **budget**)

  Second stage: Minimize the **unmet demand** for point-to-point "connections" **m**

  Each connection **m** may use different existing routes **r**

  Different routes may share one or more arcs **k**

- Uncertainty comes from: Demand for connection **m**

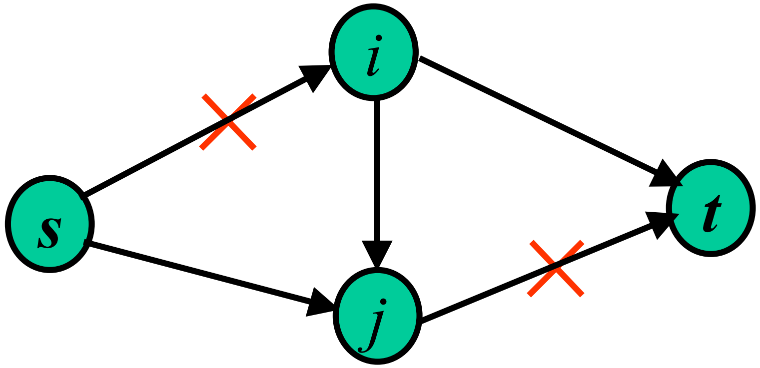
Arcs	Routes	Dems.	LB*	UB	Gap (%)	MP time	SP time
7	45	10	3.77	3.80	0.7	2 min	5 min
50	350	86	12.74	12.91	1.3	20 min	4 h
89**	620	86	9.75	10.21	4.7	4 h	30 h

(*) Prob. > 0.95 in all cases

(**) From Mak et al. (1999), Op. Res. Letters 24: (LB,UB)=(9.22, 10.06) in 43 h

Computational Results (III)

NETWORK INTERDICTION PROBLEMS (I)



MAX E\{ Min Length from s to t \}

MIN E\{ Max Flow from s to t \}

x(*i,j*) = 1 if interdiction of arc (*i,j*) is attempted, 0 otherwise

l_{ij}= Nominal Arc (*i,j*) Length (*Shortest Path problem*)

u_{ij}= Nominal Arc (*i,j*) Capacity (*Maximum Flow problem*)

r_{ij}= Amount of resource needed to attempt to interdict the Arc (*i,j*)

ξ_{ij}= Attack success for Arc (*i,j*) (Random variable):

$$\tilde{l}_{ij} = l_{ij} + \xi_{ij} d_{ij} x_{ij}$$ **("Delay" for the *Shortest Path problem*)**

$$\tilde{u}_{ij} = u_{ij}(1 - \xi_{ij} x_{ij})$$ **("Diminished" capacity for the *Max. Flow Prob.*)**

Computational Results (IV)

NETWORK INTERDICTION PROBLEMS (II)

Problem Type	Nodes	Arcs	No. allowed Interd.	LB	UB	Gap (%)	CPU	Other methods
Sh. Path	8	21	6	0.298	0.304*	2.0	1.7 min	
Sh. Path	50	893	10	137.9	142.1*	3.0	10 min	
Sh. Path	150	1,853	20	12,111	12,718*	5.0	2h	
Sh. Path	150	1,853	50	14,178	15,460*	9.0	3h	
Max. Flow	4	5	2	2.11*	2.14	1.4	7 sec	
Max. Flow	150	1,853	10	151.1*	157.4	4.1	8h	
Max. Flow**	38	67	6	10.76*	10.82	0.6	2 min	1 min
		67 ?		5.72*	5.94	4.0	**2.5 min**	6 min
		67 ?	9	3.82*	3.99	4.4	4.5 min	4 min
Max. Flow**	37	72	6	78.8*	79.82	1.3	6 min	**30 sec**
		72?		53.13*	54.95	3.4	5.5 min	8.5 min

(*) Prob. > 0.95 in all cases

(**) From Cormican et al. (1996), Op. Res. 46, No. 2

Ongoing and Future Work

- What are the actual convergence properties of the algorithm ?
- How to obtain (valid) ***M's*** as tight as possible ?
- Other representations that avoid the use of ***M's*** ?
- What helpful information might be preprocessed ? :
  - Cut dominance
  - MP with "minimized cuts" and/or "average cuts"
- What is a "good choice" for the groups sizes a priori?
- How to handle the case when LB exceeds UB?
- What additional strategies in Benders Decomp. may be used ? :
  - Elimination of inactive cuts (or Groups of cuts here)
  - Trust regions, regularized decomposition
- Less conservative strategies in terms of the probability of success
- Other linear and nonlinear representations of the estimated cuts